Special Case: Dangling Reference

Definition. Let $\sigma \in \Sigma_{DS}$ be a system state and $u \in \text{dom}(\sigma)$ an alive object of class $C$ in $\sigma$. We say $r \in \text{atr}(C)$ is a dangling reference in $u$ if and only if $r : C^0, 1$ or $r : C^*$ and $u$ refers to a non-alive object via $v$, i.e. $\sigma(u)(r) \not\subset \text{dom}(\sigma)$.

Example:
- $\sigma = \{1 \to C \mapsto \{p \mapsto \emptyset, n \mapsto \{c\}\}, c \mapsto \{p \mapsto \emptyset, n \mapsto \emptyset\}, d \mapsto \{p \mapsto \{c\}, x \mapsto 23\}\}$
- Object diagram representation:
  - $1 \to C : C p = \emptyset$
  - $5 \to C : C X = 23$
  - $1 \to D : D x = 23$

Special Case: Anonymous Objects

If the object diagram $1 \to C : C p = \emptyset : C p = \emptyset$ is considered as complete, then it denotes the set of all system states $\{1 \to C \mapsto \{p \mapsto \emptyset, n \mapsto \{c\}, c \mapsto \{p \mapsto \emptyset, n \mapsto \emptyset\}, d \mapsto \{p \mapsto \{c\}, x \mapsto 23\}\}, c \not= 1 \to C\}$.
Object Diagrams for Structural Analysis

Object Diagrams Cont'd
• dangling references
• partial vs. complete
• object diagrams at work
• Proto-OCL
• syntax, semantics
• Proto-OCL vs. OCL
• Putting It All Together: Proto-OCL vs. Software Towards Object Constraint Logic (OCL) — "Proto-OCL" —
And we assume the special unary function symbol \(\overline{\llbracket} (\_ , \_ , \_ ) \overline{\rrbracket}\), because.

That is, the following system state is forbidden in the software:

Consequently, \(\mathcal{F}_{\text{OCL}}\) is defined as follows:

\[
\mathcal{F}_{\text{OCL}} = \{ x : \mathcal{F}_{\text{OCL}}(x) \}
\]

where \(\mathcal{F}_{\text{OCL}}(x)\) is a type-consistent valuation of the logical variables.

Constraints on System States: Proto-OCL Syntax

Motivation

Example: Evaluate Formula for System State
Example: Evaluate Formula for System State

\[ \text{Formula: } \{ (\exists x A) \land (B \land C) \} \]

System State: \( x = 0, y = 1, z = 2 \)

\[ \{ (\exists x A) \land (B \land C) \} \equiv 1 \]

Conclusion: The formula evaluates to true in the given system state.
Model-driven/-based Software Engineering

Unified Modelling Language (UML)

CFA vs. Software Modelling

• Principles of Design
• Abstract data types, object orientation

VL 14

...VL 13

Putting It All Together: Proto-OCL vs. OCL

More General: Showing vs. Parsing

Where to Put OCL Constraints?
Class Diagrams can be used to graphically visualize code, define an object system structure $S$. An Object System Structure $S$ (together with a structure $D$) defines a set of system states $\Sigma_{DS}$. A System State $\sigma \in \Sigma_{DS}$ can be visualized by an object diagram. Proto-OCL constraints can be evaluated on system states. A software over $\Sigma_{DS}$ satisfies a Proto-OCL constraint $F$ if and only if $F$ evaluates to true in all system states of all the software's computation paths.

References


